

A SHORT USER MANUAL FOR THE LIQUID SURFACE DIFFRACTOMETER AT BEAMLINE BW1 IN HASYLAB AT DESY

by

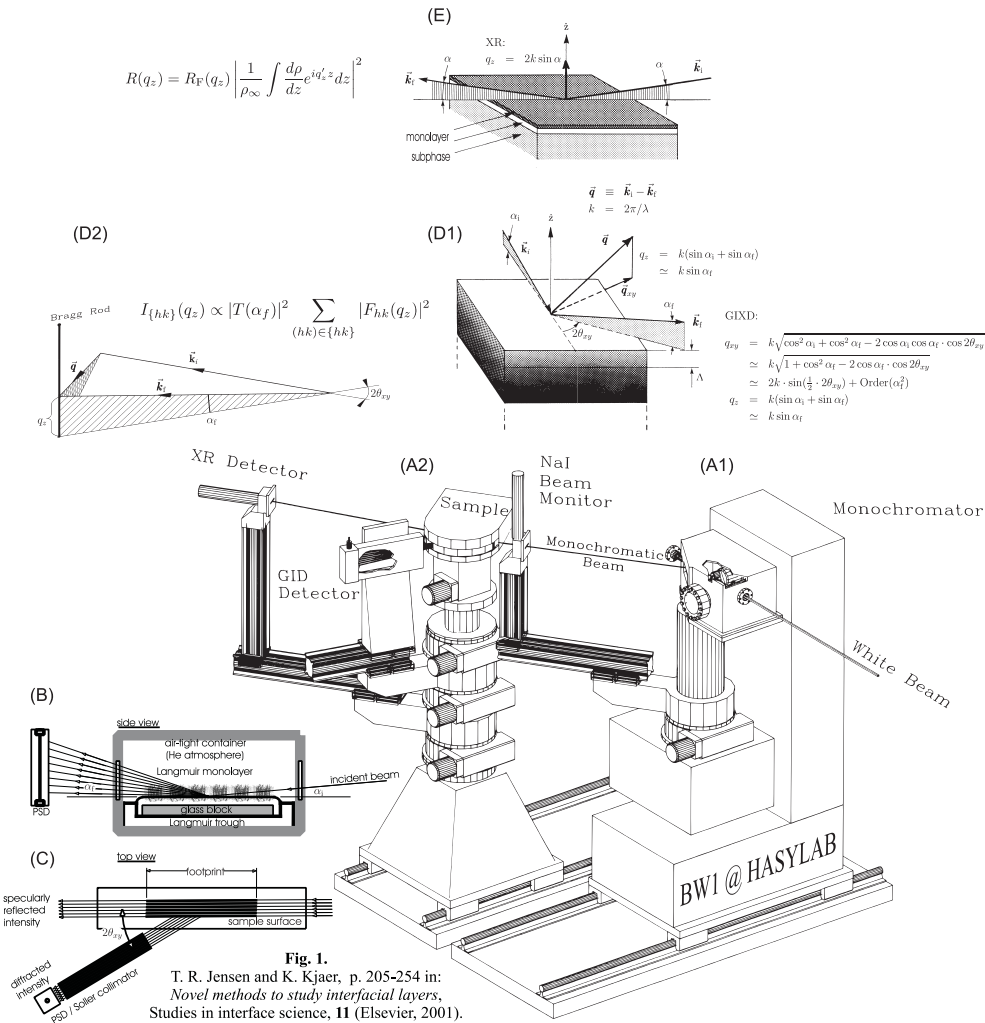
Jens Pittler <jpitt@physik.uni-leipzig.de>

with some modifications by

Kristian Kjær <kristian.kjaer@nbi.ku.dk>

University of Leipzig, Soft Matter Physics Dept., Leipzig, Germany

Niels Bohr Institute, University of Copenhagen, Denmark



Contents

1	preamble	3
2	login and startup	3
3	some useful main commands	5
4	starting a new sample	6
5	measuring an isotherm	6
6	reflectivity scan	8
7	diffraction scan	11
8	alignment	14
8.1	alignment procedures	14
8.2	sample alignment	14
8.3	alignment of the slits on the incident beam	14
8.4	alignment of the detector	15
8.5	a complete vertical alignment	17
9	description of some motors	19
9.1	movements of the horizontal scattering diffractometer	21
10	Appendix: Perl scripts by Markus Weygand	22

1 preamble

This document supplements — and to some extent duplicates — the general manual for the tascom language and associated hardware. The emphasis of the present document is on how one may go about performing a subset of the experiments that are possible by means of the BW1 liquid surface diffractometer.

Readers should first read the general tascom manual¹.

2 login and startup

The BW1 horizontal-surface diffractometer is controlled by the program TASCUM² (Triple Axis Spectrometer COMmands), which runs on a Linux system (hastasbw1.desy.de). Tascom is an interpreter of commands that are given by the user or read from *.tas files in the ~/tascom/command/main directory and specified subdirectories. Commands can be either native tascom commands (*e.g.*, an order for a motor move or an assignment to a variable) or command files. To login on **hastasbw1** use

```
Username: tascom
```

```
Password:
```

To start the program tascom use the command **tascom** if tascom is not already running. Otherwise, locate the window with the running instance of tascom. Before you do any scans you must set your own data directory. To set the data directory use

```
> data_dir='<DIRECTORY>'
```

All data filenames are ASCII files and end with **.dat** - the file names are constructed from a root string and a segment number. The variables involved are **data_dir**, **fin** and **segn**. The convention at BW1 is to use systematic names, *e.g.*,

```
> data_dir='aprjun2004/lena' fin='lena' segn=0
```

– or –

```
> data_dir='aprjun2004/lenb' fin='lenb'
```

– or –

```
> data_dir='aprjun2004/lenc' fin='lenc'
```

```
⋮
```

Thus the data file names will be composed of four letters and a four-digit number. With

```
> segn = <NUMBER>
```

¹http://www.risoe.dk/afm/external/pska/last_bw1-2.pdf

²http://www.risoe.dk/afm/external/pska/Tascom_manuals.htm

you can specify the starting number of the first file. Set `segn=0` only before the first scan in a run (as in the example given for `lena`). `SEGN` is automatically incremented with each scan. With the command

```
> dfit='<Comment>'
```

you can define a comment that is added to the header of each file. With the command

```
> log 'any text'
```

you can write 'any text' on the log printer. Set `dialog=-1`, `logp=1` and `logF=1`, cf. the `tascom` manual³.

³http://www.risoe.dk/afm/external/pska/last_bw1-2.pdf

3 some useful main commands

> shut	closes the mini shutter (to save the sample from beam damage)
> open	opens the mini shutter
> go_home	to move to the home position (<i>i. e.</i> , the same as > q1=0.85 q2=0 qtoa go_vert)
> go_refl	to move to reflection mode
> go_difr	to move to diffraction mode
> go_vert	to use the indirect (vertical) detector (in reflection mode)
> go_dir	to use the direct detector (in reflection mode)
> new_monolayer	to make a comment in the log-print, assign dfit, ... and move tls back to -15
> al1=0.8	sets the aluminium absorber (0.8 mm is the maximum value used)
> q1= 0.85 q2=0 qtoa	moves motors to incident and exit angles $in_{an} = ex_{an} = 0.85 * th_{tc}$ (= 0.1099 DEG for $\lambda_{am} = 1.3037 \text{ \AA}$, <i>i. e.</i> to a vertical $\vec{r}$ vector of length $0.85 * Q_0$
> qc= 0.021764 q0=qc	are the values for a pure water subphase.
> ? <Variable>	prints the value of Variable on the screen
> PRES = <VALUE>	assign the pre-set counting limit
> PRSC = <VALUE>	select the type of counting: prsc=0: Counting for PRES seconds prsc=1: Counting up to PRES monitor counts prsc=2: Counting up to PRES counts in the I detector (the PSD) prsc=3: Counting up to PRES counts in the AUX detector
CONTROL-C	interrupt a scan or movement

4 starting a new sample

The tascom data files begin with two lines of text. One of the text lines is meant to be a descriptive comment that can be set with

```
> dfit='comment'
> log dfit ! to log the text
```

Further,

```
> spread ! will ask for spread volume, concentration in mg/mL or Molar, and molar mass
(or just the number of molecules spread)
> lspr ! to log the spread values
> comm ! to type in a further description/comment
> tls=-15 ! to translate the sample across the beam
- or -
> new_monolayer ! to achieve all of the above
```

5 measuring an isotherm

To measure an isotherm use

```
> isot 0.5
```

You can also use `isox` to additionally record the oxygen concentration in the trough compartment at the same time, or `isotxi` to further also repeat counting. To plot the data use the programs `gnuplot` or `idl`.⁴

In `gnuplot`,

```
> pl 'lenb1234.dat' u 6:2 w l - or -
> pl '<tda lenb2345.dat AMOL PI_SURF' w l
```

In `idl`,⁵

```
idl> tapl,ctasrd('lenb1234.dat'),'Amol','Pi',psym=0 - or better -
idl> pliso,ctasrd('lenb1234.dat') - or -
```

⁴Note that these and most other linux commands must be typed in lower case. By contrast, `tascom`, `idl` and `gnuplot` will accept either case — except when referring to names of file to be supplied by the linux file system.

⁵http://www.fys.ku.dk/~krkj/IDL_stuff

```
idl> iterate, 'pliso,ctasrd('lenb1234.dat')'
```

– or –

```
idl> pliso,ctasrd('lenb1234.dat'),/hc
```

; to get a hard copy

6 reflectivity scan

To move all the motors and the detector to their home position (*i. e.*, the same as `q1=0.85 q2=0 qtoa go_vert`) (in reflection mode) use

```
> go_home
```

To save the sample from beam damage you may decide to move the trough across the beam (motor TLS) between parts of the reflectivity scan (`pref` attempts to do this automatically). For a new monolayer set `tls` back to -15. In this way you will have the maximum space for movement of the trough. To do this use

```
> tls=-15                                ! that is part of the command
> new_monolayer
```

After changing the gas in the trough cannister you need to remeasure the normalization constant with

```
> mki00.
```

One way to start a reflectivity scan is by

```
> piba
```

The macro `piba` calls the macros `pexs` – x-ray scan with indirect detector – and `pref` – x-ray scan with direct detector. The macro `pref` requires the parameters `qstart`, `qend`, `qzstep`, `spres`, `maxdosis` and `qzstepm` (see file `~/command/main/piba.tas`) where `qzstep` is the initial step width, `spres` the initial detector counts, `maxdosis` the maximum dose before moving the trough and `qzstepm` the maximum step width in `qz`. Note: q-values are expressed as multiples of `Qc`. You can change the parameters in the file `piba.tas`. To get an overview of the parameters used type `pref_para`. The command

```
> q1=0.85 q2=0 qtoa
```

calculates `inan` and `exan` according to `q1`, `q2`, calculates corresponding target values for all the motors concerned and moves the the motors accordingly. To plot the reflectivity scan (consisting of two data files in the above example) you can use `gnuplot` or `idl`.

In `gnuplot` type:

```
> pl 'lenb3456.dat' u 6:2:3 w e
> repl 'lenb3457.dat' u 12:4:5 w e
- or -
> pl '<tda lenb3456.dat q iof siof' w e
> repl '<tda lenb3457.dat q iof siof' w e
```

In **idl** type:

```
idl> pwd
idl> cd, 'lenb' ; if required
idl> lenb3456_3457 = makerefl('lenb',[3456,3457],maxrate=12000,/purge,/write)
and
idl> plotrefl, lenb3456_3457 - or -
idl> plotrefl, ctasrd('lenb3456'), ctasrd('lenb3457')
```

Other macros which you can use are

```
> go_vert m1 m2 m3 m4 m50 ! for the vertical detector
> go_dir m6 m7 m8 m9 m100 (... m110 ... m120) ! for the direct detector.
```

these record the reflectivity measurement for the following ranges:

scan	q-range	$\hat{=}$	angles	attenuator AL1
m1	0.5—1.2 * Qc	$\hat{=}$	0.06—0.16 DEG	0.8
m2	1.1—1.8 * Qc	$\hat{=}$	0.14—0.23 DEG	0.6
m3	1.4—2.2 * Qc	$\hat{=}$	0.18—0.28 DEG	0.4
m4	1.9—3.5 * Qc	$\hat{=}$	0.25—0.45 DEG	0.2
m50	3.6—5.8 * Qc	$\hat{=}$	0.47—0.75 DEG	0.0
m6	05—09 * Qc	$\hat{=}$	0.65—1.16 DEG	0.8
m7	08—15 * Qc	$\hat{=}$	1.03—1.94 DEG	0.6
m8	12—22 * Qc	$\hat{=}$	1.55—2.85 DEG	0.4
m9	17—26 * Qc	$\hat{=}$	2.20—3.37 DEG	0.2
m100	27—35 * Qc	$\hat{=}$	3.50—4.55 DEG	0.0
m110	36—42 * Qc	$\hat{=}$	4.68—5.48 DEG	0.0
m120	43—47 * Qc	$\hat{=}$	5.61—6.14 DEG	0.0

From m7 onwards a background is also measured and stored in a separate data file. After m100 you can continue with m110, m120, ... ,if you still have signal. To plot the reflectivity scan (consisting of 14 to 18 data files in the above example) you can use **gnuplot** or **idl**.

In **gnuplot**⁶ type:

```
> pl '< ct2gpl -r "lenb" -a -f -p 3456' u 1:2:3 w e
> repl '< ct2gpl -r "lenb" -a -f -p 3457' u 1:2:3 w e
> repl '< ct2gpl -r "lenb" -a -f -p -b 3458' u 1:2:3 w e
```

⁶cf. Appendix 10

```
> pl '< tasrd -s "Q1 IOF SIOF" lenb3456.dat' u 1:2:3 w e
```

In **idl** type:

```
idl> cd, 'lenb' ; if required
idl> lenb3456_3469 = makerefl('lenb',3450+[6,7,8,9,10,11],
3450+[12,13,14,15,16,17,18,19], maxrate=12000, /purge, /write) and
idl> plotrefl, lenb3456_3469
```

7 diffraction scan

To move the setup to diffraction mode

```
> go_difr
```

For diffraction measurements you may want to flush the trough with helium. Check the oxygen content in the trough with

```
> poxy
```

The command

```
> wait_oxy
```

prints the current state of the oxygen content on the screen every 10 sec. The macro will stop when the oxygen content is less than 0.3 %. Alternatively, the commands

```
> isox
> isotxi
```

– or –

record the surface pressure and area *and* the oxygen concentration in the trough container.

For the diffraction scan of 2TL⁷ there are two macros you may want to use.

```
> ttln From, To, NoOfPoints, Monitor_preset
> ttlt From, To, NoOfPoints, Monitor_preset, mm_per_100kMonitor
```

– or –

The commands execute a scan with the position sensitive detector. `ttlt` additionally translates the sample under the beam so that fresh spots of the sample are exposed to radiation. The parameters necessary to calculate the Q_z value from the channel number in the PSD file are given in the data file headers: `TH0 = * /`; `TH0` = critical angle in DEG i.e. the position of the Vineyard peak; `VCHA` = the number of the corresponding channel⁸; `SLOP` = PSD channels per DEG. These macros remove any absorber thicker than `AL1=0.2` mm from in front of the sample. To exceptionally scan with thicker absorbers, use `ttln`. Here is an example of a `ttlt`-scan:

```
> ttlt -18, -16, 81, 1e5, 0.1
```

If you want to extend a GIXD scan (or any other scan) use the command

⁷Note that all 2TL are negative numbers!

⁸given for `ExAn3=4.5` DEG, for historical reasons. Hence, for `ExAn3=6` you should observe the Vineyard peak in channel `VCHA-SLOP*1.5`

```
> reopen_f=1
```

to cause the next scan to be appended to the current data file.

At low angles you should change the horizontal width, **S3Ho**, of the slit behind the Soller collimator to reduce background scattering. The upper software limit **u2t1**, which prevents the PSD from being exposed to the direct beam, should also be increased (from -2.6 at **S3Ho**=10 to -0.5 at **S3Ho**=1). This is all taken care of by the command **set_S3Ho** that will ask for **S3Ho**. In fact, merely assigning **S3Ho**=<VALUE> is a *mistake*, cf. page 19. So doing will have no effect (other than confusing the user).

	S3HO	angle range	upper limit u2t1
> set_S3Ho 10		wide	-2.6
> set_S3Ho 3		small	-1.3
> set_S3Ho 2		small	-0.9
> set_S3Ho 1		small	-0.5

To plot the GIXD results use **idl**.

```
idl> cd, 'lenb' ; if required
idl> lenb5678=cppsd('lenb5678', update=5) - or -
idl> ppsdget, lenb5678, upd=5
```

To print the scan results use

```
idl> domp, lenb5678
idl> domp, correct(lenb5678)
idl> domp, realq(correct(lenb5678))
```

If you want to return to **reflectivity mode** type

```
> go_refl
```

Alternatively,

```
> tzss
```

can be used in-between diffraction scans. It does essentially

```
> go_refl
> saux tzs,...
> tzs=midp
> go_difr
```

8 alignment

8.1 alignment procedures

The alignment procedures consist of two parts: A scan of a motor and its subsequent repositioning. Scans are of the general type:

```
> smon Motor, From, To, NoOfPoints, PRES      ! to scan a motor and plot MON – or –
> saux Motor, From, To, NoOfPoints, PRES      ! to scan a motor and plot AUX – or –
> si    Motor, From, To, NoOfPoints, PRES      ! to scan a motor and plot I
```

Whether or not a good peak is found during the scan, a peak analysis is automatically attempted and the variables `midp` and `fwhm` contain the resulting midpoint and full-width-at-half-maximum. It is up to you to decide in each case whether the resulting values are meaningful or misleading. The command

```
> motor=midp
```

moves the motor to the so computed midpoint.

8.2 sample alignment

Do not forget to check the sample height! To scan the sample height use

```
> prsc=1 saux tzs, tzs-0.25, tzs+0.25, 26, 5000 tzs=midp
– This command is stored in function key F2. –
```

As you can see the range of this scan is calculated relative to the present motor position. It may be that a coarse alignment with the joystick is necessary first. Move to the new sample height with

```
> tzs=midp
```

Depending on circumstance, you may decide to

```
> remo tzs, sahe
```

8.3 alignment of the slits on the incident beam

The command

```
> slpo
```

writes the slit positions of incident beam SH1, SH2: the heights of the slits given by the motor positions of TZ1 and RO1; and L1, L2 the distances from the monochromator to the slits

```
> ssl1 1.5,-1.5,16,1
```

scans monitor counts (so a fixed time => PRSC=0 is used) versus the absolute vertical position of slit 1. This requires combined movement of TZ1 and RO1, so no simple motor scan can be used.

```
> ssls midp,sh2
```

moves slit 1 to the center of the peak and keeps slit 2 at its position

```
> smon tz1,tz1-1,tz1+1,21,1
```

scans the vertical height of the monochromator arm relative to its present position.

```
> tz1=midp                                ! puts slit2 in the center of the beam
> cout                                     ! counts and outputs (prints) the result
```

8.4 alignment of the detector

Alignment of the rotation of the detector arm

```
> saux ro2,ro2-0.4,ro2+0.4,17,5000
> ro2=midp cout
```

Alignment of the height of the detector arm

```
> saux tz2,tz2-1,tz2+1,21,5000 tz2=midp
- This command is stored in function key F3. -
> tz2r=midp
```

The variable tz2r now contains the actual position of the reflected beam. This is needed to determine the actual incident angle with respect to the horizontal. Moving TZ2 to the expected position of the direct beam, and the sample (TSS) 2 mm down:

```
> ptz2d
> ro2=-ro2
> cout
```

The detector now looks into the direct beam.

Alignment of the height of the detector arm:

```
> saux tz2,tz2-1,tz2+1,21,5000 tz2=midp           - Or press F3. -  
> tz2d=midp
```

The variable **tz2d** now contains the actual position of the direct beam. This, too, is needed to determine the actual incident angle with respect to the horizontal. Now move the sample back into the beam and the detector to look at the reflected beam

```
> tzs=tzs-2 tz2=tz2r ro2=-ro2 cout  
> QTRU
```

calculates from the height difference (**tz2d** - **tz2r**) and the detector to sample distance **LEN4** the actual incident 'angle' with respect to the horizontal: **QTRUE** is in units of **Qc=Q0**.

```
> qpri QTRUE,0 qpri 0.85,0
```

prints the calculated and actual motor positions for the actual incident 'angle' **QTRUE** and for the nominal incident 'angle' **Q1=0.85** and

Finally – if you are sure you know what you are doing:

```
> qcap QTRUE,0
```

This is the critical command in the alignment where the zero positions of all motors are **REMO**ed. This command expects that all motors are in their optimal positions in the reflection geometry, as determined by the previous scans. No individual motor **remo** commands are necessary. Now the diffractometer is ready for reflectivity measurements.

8.5 a complete vertical alignment

Be careful - these commands move a lot of motors. Understand what goes on, and consider how much of this needs to be done!

1. > q1=0.85 qtoa
2. > rtzm -2.5, 2.5, 21, 1 *– scan of the monochromator height⁹*
3. > ptzm midp
4. > remo tz2, t2he
5. > remo tz3, -70 *! -70 is the parkPSD position*
6. > prsc=0
7. > ssl1 1.5, -1.5, 21, 1 *– slit scan⁹*
8. > ssls MIDP,SH2 *! move sH1 to midp and leave sH2 where it is*
9. > smon tz1,tz1-1,tz1+1,21,1
10. > tz1=midp
11. > remo tz1, t1he remo ro1, tiro *! Maybe!*
12. > prsc=1
13. > saux tzs, tzs-0.25, tzs+0.25, 26, 5000 tzs=midp *– or press F2. –*
14. > remo tzs, sahe *! Maybe!*
15. > saux tz2, tz2-1, tz2+1, 21, 5000 tz2=midp ! *– or press F3. –*
16. > tz2r=midp
17. > ptz2d
18. > ro2=-ro2
19. > saux tz2, tz2-1, tz2+1, 21, 5000 *– Or press F3. –*
20. > tz2d=midp
21. > tzs=ts-2 tz2=tz2r ro2=-ro2 cout

⁹Note that here the scan limits are exceptionally given *relative* to the present positions, instead of as *absolute* positions

22. > QTRU ! – calculate QTRUE.

23. > qpri QTRUE,0 qpri 0.85,0

24. > qcap qtrue, 0 ! REMOing according to QTRUE

The QTRUE value should be equal to 0.85 ± 0.03 .

If not, do > q1=0.85 qtoa and go back to saux tzs ...

and repeat the procedure from there –

9 description of some motors

Motor movements can be directly carried out using the following syntax

```
> mot=value
```

where `mot` is a motor name from the following table and `value` is any real number. These motor positions can also be used as variables in calculations.

By contrast, assignment of non-motor variables causes no action; the value is merely stored. In the case of the variables `sH1`, `sH2`, `Q1`, `Q2`, `InAn`, `ExAn`, `ExAn3` and `S3Ho` that in fact represent motor movements, this will not be what you want. For some action (motor movement) to occur, a command must additionally be given. Examples are

```
> sH1=midp sH2=sH2                                ! Wrong!
> S1Po midp,sH2                                     ! Right!
>
> S3Ho=3                                             ! Wrong!
> set_S3Ho 3                                         ! Right!
>
> ExAn3=12                                           ! Wrong!
> ARDX 12                                           ! Right!
>
> Q1=0.85 Q2=0                                       ! A mistake, most probably!
> Q1=0.85 Q2=0 QtoA                                 ! Right!
>
> InAn=0.109964 ExAn=InAn !                         A mistake, most probably!
> InAn=0.109964 ExAn=InAn AtoQ !                   Right!
```

The motor `AL1` (aluminium absorber wheel) is in a sense a special case: Assignments such as

```
> AL1=0.8                                           - or -
> AL1=0.0
```

cause the motor to move to the expected aluminium absorber thickness (in mm). (Such is the calibration of this motor.)

However, as inspection of the device should convince the reader, assignments such as

```
> AL1=0.75                                           - or -
> AL1=-0.1
```

will *not* give the corresponding absorber thicknesses.

Concerning AL1 (though completely unrelated to the issues discussed above), note that working with the trough one can easily happen to offset the rotation of AL1 since the holding torque of the motor is weak. Only visual inspection of the orientation of the aluminium absorber wheel will reveal such an offset.

9.1 movements of the horizontal scattering diffractometer

Function	Tascom	Unit	Description	Location
TF	TFM	mm	Horizontal Translation @ Floor	M
TZ	TZM	mm	Vertical Translation	M
HA	HAM	Deg.	Horizontal Aperture	M
VA	VAM	Deg.	Vertical Aperture	M
	OMM	Deg.	Crystal Rotation	M
G ()	GM	Deg.	Crystal tilt (Goniometer)	M
	2TM	Deg.	Horizontal arm Rotation	M-S
	INAN	Deg.	Vertical Incidence Angle to Sample	M-S
	RO1	Deg.	PI arm Rotation	M-S
	TZ1	mm	PI arm Vertical Translation	M-S
sH1	sH1	mm	Vertical positions of slit 1	M-S
sH2	sH2	mm	Vertical positions of slit 2	M-S
TF	TFS	mm	Horizontal Translation @ Floor	S
	OMS	Deg.	Sample Rotation	S
TZ	TZS	mm	Vertical Translation	S
TU	TUS	mm	Upper Translation (not motorized)	S
TL	TLS	mm	Lower Translation	S
GU	GUS	mm	Upper Goniometer arc (not present)	S
GL	GLS	mm	Lower Goniometer arc (not present)	S
φ	2TL	Deg.	Horizontal Lower arm Rotation	S-PSD
ψ	2TU	Deg.	Horizontal Upper arm Rotation	S-Aux
	EXAN	Deg.	Vertical Exit Angle from Sample	S-Aux
	RO2	Deg.	PI arm Rotation	
	TZ2	mm	PI arm Vertical Translation	
	EXAN3	Deg.	Vertical angle from Sample	S-PSD
	RO3	Deg.	to middle of PSD	
	TZ3	mm		
/	Q1	–	$(\varphi) \cdot (\psi + \varphi) /$	M-S-Aux
/	Q2	–	$(\varphi) \cdot (\psi - \varphi) /$	M-S-Aux
Atten	AL1	mm Alu.	Alu. atten. before beam monitor	M-S
Atten	AL2	mm Alu.	Additional Alu. atten. (not used in the present set-up)	

) M = monochromator; S = sample; Aux, PSD are detectors.

) Not a motor: Movement implemented by means of motors RO1, TZ1, resp. RO2, TZ2 resp. RO3, TZ3

10 Appendix: Perl scripts by Markus Weygand

The perlscript `ct2gpl` helps to deal with reflectivity data inside or outside gnuplot. In a shell it prints the result to standart output, that is your terminal and you may pipe to other tools or directly into a file you want to generate. It wants the prefix of your current data names e.g. `lenb`, otherwise it takes a default one which is likely not the one you are currently using. Additionally you have to supply at least one number of a data file that contains reflectivity data e.g. 3456. You may provide two — in this case `ct2gpl` regards the second number as the last file in a sequence of reflectivity mesasurents the containing reflectivity data (not the background data, which is usually the file following).

If you must subtract backround from your data file you have to use the option `-b` and `ct2gpl` calculates background corrected data. In the case there are fewer background data points than data point it does linerly interpolate background data points.

Use the option `-p` to cut out data points, that are measured too fast by the detector.

Sometimes you may not have reflectivity and background data as two files in a sequence and thus want specifically to specify data and background file numbers. This is easily done by using the option `-m` and providing the data file and background number.

Furthermore option `-a` preserves units of critical angle and `-f` normalizes output to fresnel reflectivity. For an overview of more options you may type `ct2gpl -h`.

In a more general context another perlhack `tasrd` provides you save acces (`ct2gpl` is also save, i.e. it does not rely on a specific column number where to fetch its input) to tascom data file columns. Lets say if you want to look just in a reflectivty data file you may type `tasrd -s "Q1 REFL SREF" wolb0764.dat` or looking at a isotherm you are recording by TASCOS you may use `tasrd -s "AMOL PI_SURF" wolb0440.dat`. You can access header parameter instead of data columns by choosing the option `-p`, fx. `tasrd -s "TEMP" -p wolb0764.dat` gives you the temperature TASCOS recorded during the scan.